The OneSAF Testbed Baseline SAF Puts Added Simulation Capabilities Into Users' Hands

Pamela Bowers
CrossTalk

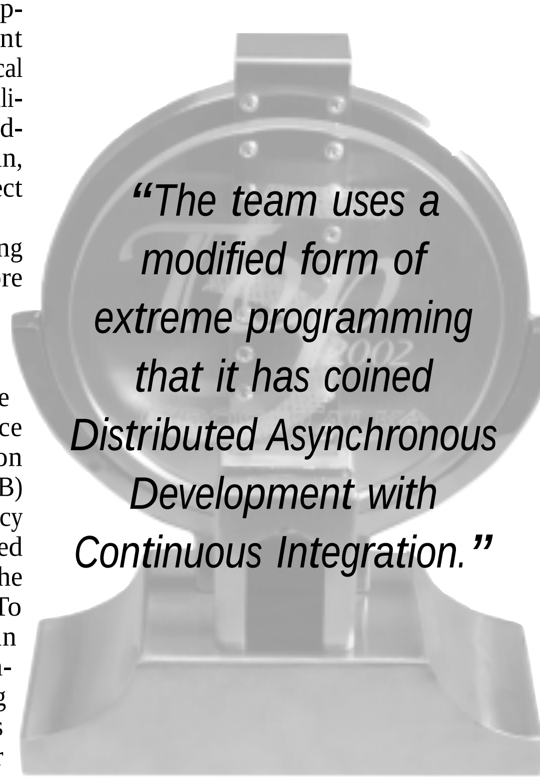
The U.S. Army has used semi-automated forces (SAF) for training, analysis, and research since the mid-1980s. The OneSAF Testbed Baseline SAF (OTB SAF) serves as a bridge between the legacy SAF system and the OneSAF Objective System[1]. OTB SAF is an open source solution that users can configure to meet their needs.

Simulation has long been used as a military training tool for land, water, and air combat. The U.S. Army has used semi-automated forces (SAF) for training, analysis, and research since the mid-1980s. These SAF accurately and effectively represent the physical behavior of joint weapons systems as well as the tactical behavior of individual entities and military units. They also portray detailed models of the natural environment (terrain, atmosphere) and the environmental effect on simulated activities and behaviors.

These SAF were in need of updating to allow military training to reflect more modern-day warfare, terrain, and resulting effects on the warfighters, as well as support analysis and research on developing Army platforms. The software program maintained by Science Application's International Corporation (SAIC), OneSAF Testbed Baseline (OTB) SAF serves as a bridge between the legacy SAF system – Modular Semi-Automated Forces (ModSAF) Version 5.0 – and the OneSAF Objective System (OOS). To accomplish this, SAIC developed an open-source solution that maintains configuration management of existing ModSAF capabilities and enhances these capabilities to support interim user requirements. In addition, OTB seeks to reduce risk for OOS development by providing opportunities for integration, test, and user feedback on technology developments.

"The key challenge in this project," says Bryan Cole, SAIC division manager of Simulation and Training Systems, "was to translate the needs of the go-to-war soldier into software requirements that would result in a product that provides the capability the soldiers were looking for, and was sufficiently user friendly."

OTB is used at more than 220 U.S. sites and several international locations. It can be used as a stand-alone simulation, or as an embedded system within a manned simulator. It can also interact in a joint exercise with other live, virtual, and constructive simulations using the Distributed Interactive Simulation and/or High Level Architecture (HLA) simulation standard.



> *"The team uses a modified form of extreme programming that it has coined Distributed Asynchronous Development with Continuous Integration."*

The OTB empowers trainers, analysts, and researchers to configure the simulation to meet their needs without total reliance on software developers. Each version of the OTB puts more and more power into the hands of the users, allowing them to tailor the application for specific requirements.

OTB operates as a distributed system, though the current architecture supports interface to servers (e.g. weapons effects) if desired. Typically, there is no real client or server in the architecture. Workstations negotiate load balancing, and the distributed nature of the application allows recovery from individual system crashes without interruption to the simulation scenario in progress. Methods also exist to participate in a simulation using a distributed network architecture. OTB is easily configured using simple text files and can be modified in the field without needing to be re-compiled from the source code.

OTB 1.0 represents a major overhaul of ModSAF 5.0 code, including the removal of non-functioning libraries, the enhancement of outdated algorithms, implementation of a native HLA interface and the implementation of major new SAF functionality. The update impacted nearly all of the existing ModSAF 5.0 libraries.

Development Method

The Orlando, Fla.-based SAIC group is a Software Engineering Institute Capability Maturity Model Integration (CMMI®) Level 4 organization. This accomplishment and the quality of team members and a progressive customer (Program Executive Office – Simulation, Training and Instrumentation) are what Cole attributes to the project's success. "They have years of disciplined experience applying process to the software development undertaking," he says. "We use basic CMMI processes. Given the nature of the application, it's very complex. However, since its open nature provides a fair amount of latitude to work within the specified requirements to come up with a solution, we encourage creativity and creative thought."

The primary development platform for OTB is a Linux workstation using Debian Linux Version 2.2. A typical hardware configuration is a 1GHz Pentium III with 512 MB RAM and a 60 gig hard drive. Efforts are nearly complete to allow migration of the development platform to Red Hat Linux Version 8.0.

Due to the large installed base of ModSAF users, OTB dictated support of a wide number of operating systems. These include Silicon Graphics IRIX 5.3, 6.2, and 6.5; Sun Solaris 2.5.1 and 2.6; current Debian; Red Hat Linux; and Windows NT. The development team conducts specific tests of new code on

® CMMI is registered in the U.S. Patent and Trademark Office.

each operating system. OTB runs on a wide variety of workstation hardware. Using a minimally configured Linux workstation based on commercially available PC architecture (e.g., 900MHz PIII, 256MB, 4GB HD), OTB can simulate 200 or more individual combat entities per workstation.

The team used spiral or incremental development or a hybrid of the two at various times depending on the effort, says Cole. Multiple developers are on different time lines at any given time depending on the requirements of the particular task, and when it will ultimately be integrated into the baseline. Commercial off-the-shelf products comprise the visual system.

OTB currently consists of more than 1.3 million lines of code in the entire baseline, excluding comments[2]. The source code includes 592 individual libraries. The compiled binary and reader files can be installed as a minimal configuration in as little as 30MB, but the source code tree is more than 135MB prior to compilation.

The team uses a modified form of extreme programming that it has coined Distributed Asynchronous Development with Continuous Integration, says Cole. "The team works in pairs and follows a *build a little, test a little* pattern of incremental development. There are multiple developers addressing requirements at different times during different time lines, requiring support for continuous integration," he says. "This works well at the end because you don't have this big-bang integration."

Long time programmers are typically paired with inexperienced programmers unless there is a specific reason to do otherwise. Recently, Cole says that actual go-to-war operators sat down with SAIC programmers to provide over-the-shoulder feedback. "These guys were the subject matter experts. It's fairly well acknowledged that you get the best product when you have the user right there in the development environment with you. That certainly was true in this case."

The team prides itself on its ability to work in that integrated product team environment. "The concept of a collocated team works," Cole says. The customer, end user, and developer sit down and understand the requirements. Then that understanding is used to develop an incremental build with the user standing over the team's shoulders during testing. "Some of the bedrock of our success is that integrated relationship with the customer, end user, and developers all working according to a common understanding."

The Science Application's International Corporation's OneSAF team gave users configuration power.

Quality In, Bugs Out

As part of its CMMI Level 4 processes, SAIC does an extensive amount of internal peer reviews. Developers use Concurrent Versioning System software to actively track and control code integration and release configurations. All coding changes are peer reviewed, checked by developers, submitted to a single point integrator for inclusion in the developmental tree, and submitted to a separate test group for regression testing and functionality verification.

Once the development team is satisfied, the build goes to the customer or user. "In this way, we may see some changes during the design phase, but we're no longer seeing design changes while we're testing," says Cole. "We get the design stable during the design phase through peer and customer reviews. Once we have the design articulated to the customer, and they acknowledge it will satisfy their requirements, we don't see any changes typically after that point. So we spend a fair amount of time getting that out of the way on the front end."

Cole credits being a Level 4 organization with this accomplishment. "Metrics that we collect concur that we're seeing fewer and fewer errors downstream in the processes as a result of the focus on the front end, in the analysis and design stage, and with peer review. As everybody knows that makes for less expensive software and more predictable schedules." SAIC has experienced a notable reduction in production costs and consistently satisfies a diverse customer base, ranging from development of a Comanche helicopter model to the support of a large scale exercise at Ft. Knox.

Government and contractor managers conduct regular status checks of project progress from both engineering and programmatic perspectives. These reviews are conducted in an Integrated Product Team environment, and are reinforced by routine contractor program management and quality assurance assessments.

"We pride ourselves on on-time delivery of a fully functional baseline," says Cole. "We manage the development process very, very closely using an earned value system so we actually plan what we'll accomplish for each build so we can measure that value and progress as we go. If it looks like something is slipping, we can add additional resources to get caught back up before the ultimate delivery date."◆

Notes

1. An objective system is the term applied to a system that has existence or authority, apart from any individual's experience or thought; in other words, the OneSAF Objective System will stand because of its technical merit.
2. This count was generated using a customized script that SAIC engineers built to automate software-line-of-code (SLOC) counts for the OTB software. This tool counts a full or partial line of executable code as a SLOC, according to industry standard criteria.

Project Point of Contact

Bryan Cole
Science Application's International Corporation
Division Manger of Simulation and Training Systems
Phone: (407) 243-3346
E-mail: bryan.a.cole@saic.com